



STOP PRESS

DRAGON

Voice of the Dragon

The Dragon 32 home computer was launched in August 1982. Since that date it has become the best selling home computer in its price bracket. There are now over 40 000 Dragon users in the United Kingdom and by the end of 1983 this figure is expected to exceed 150 000.

This represents an enormous number of Dragon enthusiasts around the country who are enjoying the power and versatility of the Dragon 32.

To help keep Dragon users informed of the very latest developments in Dragon technology, including the introduction of new peripherals, the latest on software releases and even the development of new Dragon computers, an official newsletter will be sent directly to Dragon users quarterly in 1983.

Aimed primarily at existing Dragon 32 owners, the newsletter will be used to create a forum for articles on the many facets of programming the Dragon.

If you would like to contribute programs, comments, suggestions, hints or stories on using the Dragon 32 then please write enclosing details to:

The Editor, The Newsletter, c/o Dragon Data Ltd., Kerrig Industrial Estate, Margam, Port Talbot, West Glamorgan

Submission of Programs

All program submissions should enclose a program listing and a letter explaining its basic structure. Long programs should

be submitted on a cassette tape which is itself clearly labeled with your name and address, together with a self-addressed envelope for its return. List program lines with a maximum of 64 characters per line (two screen lines) and number lines in multiples of 10.

The Dragon Users Club

To help everyone obtain the maximum enjoyment and benefit from their Dragon 32, the newsletter will pay specific attention to the formation of a Dragon Users Club.

A special feature in each magazine will be devoted to club news, giving details of specific events of interest to Dragon users, both at a local and national level.

Details of how to apply for membership of the Dragon Users

Club will appear in the next edition of the newsletter which will be mailed free of charge to all existing users who have returned their guarantee cards for the 32 to Dragon Data Ltd.

Dragon Users

The newsletter will also report on the very latest developments happening within Dragon Data Ltd., the Welsh based manufacturing company that produces the highly successful Dragon 32.

1983 will see some exciting product news for Dragon users. Current plans include the introduction of a disk drive for the Dragon 32, scheduled to appear on the market in April.

For the latest news on software plans and for an up-to-date list of software currently available for the Dragon 32 see the back page of this newsletter.

Cassette Loading

Cassette loading and saving with the Dragon is generally straight forward but problems may be encountered with some Automatic Level Controls on cassette recorders.

If problems do occur with saving and loading your own programs onto cassette here are some tips that may be of use.

1. Check that cassette heads are clean, and clean the heads regularly.
2. Ensure that plastic tape header has been passed before saving.
3. If IQ errors are obtained and television is very close to Dragon, move Dragon further away from the television.
4. If problems still occur try removing input lead (MIC) whilst loading and output lead (EAP) whilst saving.
5. The problems caused by some Automatic Recording levels may be cured by using the following command (Put Motor on first):
SOUND 120, 20:FOR N=1 TO 100:NEXT C:SAVE "Program Name"

Programs f



DRAGON BYTES

Although DRAGON has 32 768 bytes of memory, which should be ample for most purposes, some of us always want to squeeze the last byte out of the machine. If you are embarking on one of those large multi-option programs, a few tips may help to put off the moment when the CM ERROR appears and you're in trouble.

REM statements are very useful in developing and editing programs, but every letter takes 1 byte, and so these should be deleted if memory is running short. But be careful not to delete whole lines if the line numbers are used in GOTO or GOSUB statements.

Most spaces are unnecessary, and each takes 1 byte. But some of them must be kept in. Apart from the obvious ones (break, quote, and a space in C/C++ statements), a space is

needed when a variable name is followed immediately by a word. In the following examples the spaces are necessary:

```

VAR100 = 1000  VALUE = 1000000000  STOP  GO  GO  GO  GO  GO  GO  GO

```

These spaces claim, in fact, to be removed, but only by means of special programs, usually in machine code. The advantage of such programs is that you can keep a "readable" version and a "compact" version of your BASIC programs — without spaces and `REM` statements, it's not easy to modify them.

Use 1 character variable names, a preference to 3 character ones, and never more than 3.

Finally, whenever possible, write several statements on one line. Writing a statement on a new line takes 4 bytes more than adding to a list statement line.

With 32K floppy with you, you'll often need to worry about this sort of byte budgeting. But it pays to develop economical habits—and it's more satisfying to program efficiently.



SKETCH PAD

The program uses the high-resolution graphics screen to draw pictures. The target cursor can be switched on and off using the space bar; left and motion using the arrow keys; and stopped using the full stop (Shift with the arrow keys gives a 40° rotation). With a bit of practice you will soon learn to draw pictures with it. Then you might like to try modifying the program to use colours (it's a beer-resolution screen) and to make use of the PAINT command. If you have joystick(s), you can also convert the command to joystick(s) stuff.

[illegible]

TYPING GAME

This is a **free** program to help you find your way around the keyboard. It uses **THINK** to check when you have typed the letter, and the **THINK** to help you a second.

```

1  / TYPING GAME
2  / MARKS READING TIME
10  CLS PRINT "TYPING GAME" : PRINT "2" = 0
20  PRINT "THIS PROGRAM ASKS YOU TO FIND
30  PRINT "A LETTER IN THE KEYBOARD AND"
40  PRINT "HOW MANY TIMES IT TAKES YOU TO TELL"
50  PRINT "YOUR OWN LOGIC YOU TAKE"
60  FOR I = 1 TO 10000 NEXT CLS
80  FOR K = 1 TO 10 : C = RND(26) + 65 : TIME = 0
90  PRINT "1) SET TYPE THE LETTER :"; C; PRINT "COUNT 100 X"
100 : C = RND(26) : IF C = " " THEN 70
90  PRINT "2) PRINT"
60  C = C - 32 : IF C = CHR(0) THEN 110
110  PRINT "SET THE NUMBER = PROBABLY 0 OR 1"
120  TIME = TIME + 1
130  PRINT "TIME"
140  PRINT "REACTING TIME :"; INT(1/10 * TIME) / 1000 ; " SECONDS"
150  FOR I = 1 TO 10000 NEXT PRINT "2" = 1 : 1 CLS NEXT K
160  PRINT "AVERAGE REACTING TIME COUNT"
170  PRINT "IN 10000 X"; INT(1/10 * 1000 * 1000) ; " SEC"
180  PRINT "HOW ABOUT ANOTHER 1000 X"
190  TIME = 0 : IF C = " " THEN 170
200  TIME = 0 : CLS : FOR I = 1 TO 10000 NEXT
210  TIME = " " THEN CLS : GOTO 100
220  END

```

If you wish to learn to type, there is a Dragon type assistant called "Learn Type".



WHAT ARE HEXADECIMAL NUMBERS?

All the heart of your Credit is a reflection of the credit you give. So, give it a solid credit. Call 1-800-222-2222.

bits consists of 8 locations, which can be either 1 or 0, forming a binary number between decimal 0 and decimal 255. Binary numbers are very difficult to deal with without making mistakes, and the 256 numbers can be represented by two digits, each of which can be the symbols 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A for 10, B for 11, C for 12, D for 13, E for 14 and F for 15. Near Oregon, this is built in function 44252. No which converts the decimal number M to its

nonacademic equivalent. But you might like to follow through the small process which performs the same operation.

```

10 INPUT "ENTER A POSITIVE INTEGER: ";
20 N = " " : S = 0
30 IF N < 1 THEN GOTO 40
40 S2 = S + 1 : S = S + N : N = N - 1 : IF N > 0 THEN GOTO 30
50 S = S / 2 : GOTO 60
60 PRINT "N = " : PRINT "S = "
70 IF N < 1 THEN GOTO 80 : N = N + 1 : GOTO 30
80 END : PRINT "END OF PROGRAM"

```



MACHINE CODE CORNER

Machine code is the language your Dragon understands. It can only communicate in languages like BASIC by means of an interpreter, and translating languages takes time — even for a Dragon. So if you write your instructions in machine code, they will be carried out faster, and in some cases you will be able to do things which was not possible in BASIC.

At first sight, machine code doesn't look much like a language at all. It has no words — just a series of numbers or bytes. Each byte lies between 0 and 255 (8 bits) and represents an instruction, or part of an instruction. The Dragon executes these instructions in order in much the same way as a BASIC program is executed. A long and complex machine code program is a permanent feature of your Dragon — it is the BASIC interpreter and a hold in ROM (Read-Only Memory) between decimal addresses 32768 and 48191 (hexadecimal 8000-9FFF) (for more information on hexadecimal numbers see opposite.) To look at a small part of this program, try running the following Basic program.

```
10 FOR I=48000 TO 48147
20 PRINT PEEK(I); NEXT I
```

Then typed out what the machine code program does: type EXEC48000.

(All commands of the cart must, of course, be followed by ENTER.)

By the way, it is not a good idea to EXEC48000 if you have an important program in memory, since one of the results is the same as typing "Halt".

Because of Dragon's memory is ROM, and as it can't be altered. When we write our own machine code programs, we must locate them in RAM (Random Access Memory), between decimal addresses 0 and 32767 (hex 0-7FFF). All of them can be altered using the POKE command, but some of them are used by the system for special purposes. Of particular interest to us is the range 3284-1325 (hex 800-9FF) which is the Text Screen Memory. Altering these locations causes characters to appear on the Text Screen.

A good place to locate machine code programs is in the highest RAM locations (say 32000-32767) since this region can be "protected" from BASIC using the CLEAR command.

The easiest way to produce machine code is by writing in assembly language, which is a more concise representation of the code, and using an assembler to assemble the machine code. Two Editor Assembler packages are available from Dragon Data Limited. An assembler is essentially equipment for writing anything but the simplest machine code routines.

The following is an example of a simple program to fill the Text Screen with the character whose ASCII code is stored in the highest RAM address (hex 7FFF).

Assembly Language	Description	Machine Code
1 LDA 3FFF Load A		187 137 255
2 LDR 3400 Load R		142 4 0
3 LOOP STA A+ Store A		187 128
4 CmpR 8000 Compare R		140 8 5
5 BRG LOOP Branch if reg. eq		35 249
6 RTS Return from subprogram		57

Line 1 loads register A with the contents of memory 32767 (hex 7FFF). Line 2 loads register R with the value hex 400 (the address of the top left corner of the Text Screen). Line 3 stores A in the memory indicated by R and increments R by 1. Line 4 compares A with the value hex 800 and line 5 causes a branch to line 3 if A is not equal to hex 800. Line 6 returns control to the calling program.

As you can see, statements in this program take 1,2 or 3 bytes when translated into machine code. The whole program becomes:

```
187 137 255 142 4 0 187 128 140 8 5 35 249 57
```

To load this program, without using an assembler, you must write a short BASIC program:

```
10 CLEAR 320,32000
20 FOR I=2 TO 13:READ J:POKE I+32000,J: NEXT I
30 DATA 187,255,142,4,0,187,128,140,8,5,35,249,57
```

This will "POKE" the machine code directly into the memory starting at 32000, having first protected that area with the CLEAR command. Before running, it is wise to check very carefully that it has been copied correctly. Unlike BASIC programs, machine code routines will not return control to you with an "error" message if you make a mistake. Sometimes the Dragon will just refuse to respond no matter what keys you press. Should this happen, control may be restored by pressing the RESET button, but it is generally best to switch off and start again, but don't worry — you can't damage your Dragon, no matter what you POKE into its memories. It will always go itself right if you switch off and start again. So you can "experiment" as much as you like.

Now we have written and loaded our machine code program, it is time to see what it can do. It is best to run it under the control of a BASIC program, such as the following:

```
10 CLEAR 200,32000:CLS
20 G=INKEY$:IF G=" " THEN 30
30 POKE32767,ASC(G)
40 EXEC32000:GOTO 30
```

Run this program, and by pressing different keys, the screen should fill with the relevant character instantaneously. You may like to try to achieve this result using BASIC only, and see how fast you can fill the screen. The difference between letters and other symbols is interesting. By pressing (Shift) keys and see what effect this has on subsequent keys.

Now obtain control by pressing the BREAK key and change line 30 to read 30 POKE32767,ASC(G)*4 and run this new program. The symbol keys will now give a "spray effect", but the letter keys still work — in machine code.

Now replace line 30 with 30 POKE32767,ASC(G)+128

The result is an interesting display of graphics. If you would like to save your machine code on tape, the command route is SAVE(M "SCREEN", 32000, 3200) to ASCII-R. Symbols replaced by any filename you choose, the first two numbers are the addresses of the beginning and end of the machine code, and the third number is the difference between them. All of the parameters are necessary. The machine code may now be loaded from tape using LOAD(M "SCREEN"). The BASIC program could be saved in the usual way (SAVE).



DRAGON SOFTWARE

On the subject of adventure games, *Mistwood* and the *Mindbogg* seems to be living up to its name. As we have had a number of inquiries concerning the *Mindbogg*, we are taking the opportunity to give you a few more clues:

1. The mushroom can be found two floors down from the entry floor.
2. To descend you will require the lamp. To switch the lamp on, enter the command "LAMP ON".
3. On the floor where the mushroom is to be found, you stand

The range of software available for the Dragon 32 will be increasing all the time. More games on cassette and cartridge are scheduled for release over the next few weeks, including a series of new cassettes for "adventure fans".

On the subject of adventure games, *Mistwood* and the *Mindbogg* seems to be living up to its name. As we have had a number of inquiries concerning the *Mindbogg*, we are taking the opportunity to give you a few more clues:

A good chance of getting lost in a seemingly endless maze (looking for South and later East as you can) should get you out of that.

The first three of a series of ten educational cassettes should be available by the time you receive this newsletter. The other seven will become available over the next six months. The programs are designed to assist the numeracy and literacy development of four to twelve year olds. The first titles are: —

1. **Hide and Seek** Designed to encourage short term memory and aid development of early reading skills.
2. **Number 1 Puzzle** Designed to improve children's ability to do mental arithmetic.
3. **Number Guesser** Another cassette designed to improve arithmetic which children will enjoy using.

SOFTWARE AVAILABLE FOR THE DRAGON 32

A 0502 DRAGON SELECTION ONE

Four games for the younger user. *Wormy*, *80/80C*, *Flay* can be loaded and edited.

A 0501 DRAGON SELECTION TWO

Collection of utilities. Create your own data base, write your own tunes.

A 0502 QUEST

Adventure game in a medieval setting. Defeat *Mistwood*, master of the dark castle.

A 0503 MINDBOGGS AND THE MINDBOGGS

A real time adult adventure game.

A 0504 PERSONAL FINANCE

Keep track of family finances.

A 0505 GRAPHICS ANIMATION

Create simple cartoons on the screen and animate them by flipping through the pages. Joysticks required.

A 0506 COMPUTABOGE

Your Dragon will talk with this speech synthesizer.

A 0507 EXAMPLES FROM THE MANUAL

30 examples from the programming manual.

A 0508 GALTYS ISLAND

An adventure game. Return the hidden treasure to its rightful place.

A 0509 BLACK SANCTUM

An Adventure game. Overcome the forces of black magic.

A 0510 TYPING TUTOR

Improve your speed and accuracy.

A 0512 DRAGON MOUNTAIN

An adventure game. Defeat the guardians of the treasure hidden in the mountain.

A 0514 FLAG

Race your opponent through a constantly changing maze to the final flag. Joysticks required.

A 0516 EL CHARLENO

An Adventure game set in the desert.

A 0100 BERSERK

A challenging shooting game, based on the popular arcade game, one or two players. A high resolution game in black and white. Joysticks required.

A 0101 METEORITES

Guide your ship through treacherous asteroid belt. A game requiring skill, fast reactions and concentration. A high resolution game in black and white. Joysticks optional.

A 0102 COSMIC INVADERS

Dragon version of the famous arcade game.

A 0103 GHOST ATTACK

Next game for one player. Joysticks required.

A 0104 CAVE HUNTER

Descend into the maze of caves in search of gold. Joysticks required.

A 0105 STARSHIP CRASHLORD

Protect your planet from the attacks of the Galaxians. High quality arcade game with superb graphics and sound. Joysticks required.

A 0107 ASTROBLAST

Defend your ship against waves of attackers. A high resolution game in black and white. Joysticks required.

A 0108 CHESS

Nine levels of play from beginner to master.

A 0111 RAIL RUNNER

Move Bill Switchman across the tracks, avoiding trains, to rescue Herman Hobbs.